Merging line segments in 3D using mean shift algorithm in man-made environment *

Robert Schwarz, Sven Olufs, Markus Vincze

Automation and Control Institute
Vienna University of Technology, Austria
www.acin.tuwien.ac.at
$\{$*schwarz,olufs,vincze*$\}$*@acin.tuwien.ac.at*

Abstract
In this paper, we present a novel approach that is able to cope with fragmented lines by post processing the 3D lines in euclidean 3D space based on the mean shift algorithm. Starting with mapped line segments from a stereo camera setup, the lines are grouped by their direction and their proximity, so small parts of a big line structure merge into only one. This step provides not only accurate information, but also a reduction of data for subsequent processing.

1. Introduction

Stereo vision 3D reconstruction is an important task for visual perception.In recent years the research focused on points, e.g. KLT features, or regions of interests, e.g. superpixel approaches, while line segments were mostly unattended. Many line based methods depend on proper line detection of the input 2D image pair i.e. neither lines are fragmented nor able to cope with partial occlusion. Our approach makes no restrictions to the line detector and deals with the problem in the 3D space. The flawed line segments are grouped by a mean shift algorithm. This algorithm clusters perceptually similar lines from the set of potentially fragmented 3D lines and combines lines from a cluster to one instance. No preprocessing of lines in the image space is required.

This paper is organized as follows: Section 2. presents the related work, section 3. gives a brief introduction on the mean-standard deviation line descriptor. In Section 4. the focus is on the merging process with mean-shift clustering, and is followed up by Section 5., which demonstrates the experimental results. In the final section, we give a short conclusion and a brief outlook on future improvements.

2. Related Work

Literature proposes various approaches for 3D reconstruction, depending on the sensor used. For stereo based reconstruction the multiview geometry approach by Hartley and Zisserman [3] is prevalent. Independent from the number of views, one of the main processing steps is to match features. This features can be geometric objects like points, lines, and planes. There are many approaches for point features based on SIFT descriptors [6], but less for lines. Zisserman proposed in [10] an auto-

*The research leading to these results has received funding from the European Community's Sixth Framework Programme (FP6/2003-2006) under grant agreement no FP6-2006-IST-6-045350 (robotshome)

matic line matching for epipolar geometry, but does not explicitly deal with fragmented line segments. The work from Neubert and Protzel [9] also deals with line features, global and local extraction of lines used to track lines in real-time without knowledge of camera motion. Another approach deals with matched lines linked with SLAM; the development of visual simultaneous localization and mapping. In [11] lines are parametrized with Pluecker coordinates for an undelayed initialization of the line segments, however the lines are not fragmented.

Our work deals with these fragmented lines and the techniques used in our approach fall into several categories: line detection in single images, merging lines, and 3D reconstruction. To begin with, line detection is a classical problem in computer vision, and there are several ways of getting lines from images. The typical strategy can be divided into three steps: first, the edge detection, followed by chaining the edges and fitting the lines.A good comparison of line detectors is shown in [8]. Apart from Hough-transformation relatives, [4] provides an approach for grouping pixels with similar gradient orientation which was implemented in our work.

The problem of fragmented lines is processed in [5] for 2D maps in robotics. A set of line segments extracted from laser range scans are merged by using mean-shift clustering. Our attention focused on the problem of broken line segments in 3D space, based on the mean shift algorithm the merging process was done without any preprocessing steps in the image space.

3. Mean-standard Deviation Line Descriptor for Matching

This section shows how lines from two images (e.g. stereo camera setup) can be matched. In comparison to [10] our approach for matching this segments from left and right image, are not only based on epipolar geometry, but also on a simple adaption of [12]. The problem with only epipolar view is among other things the inaccuracy of line endpoint locations. Therefore, line segments from one image can not be matched to associated broken or overlapped line segments from the other image without any manipulation of the lines in 2D.

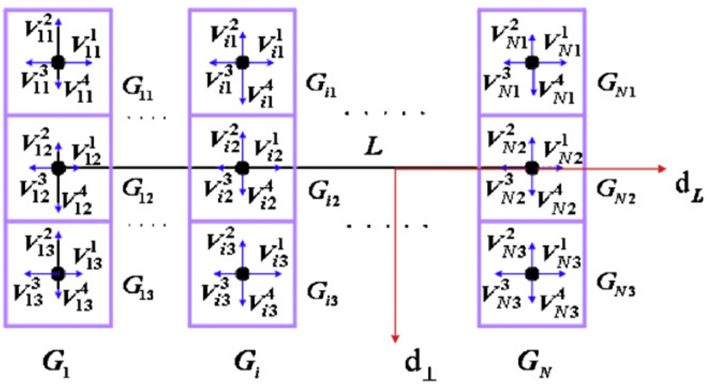


Figure 1. A schematic figure of MSLD construction with three sub-regions (taken from Wang et. al.[12])

Our matching process uses a SIFT-like line descriptor [12]. Fig.1 shows the construction of the mean-standard deviation line descriptor (MSLD) for a line L. Two directions are defined, d_L in line direction and $d_\perp$ in orthogonal direction. For every pixel on the line segment a pixel support region $G_1, G_2, ...G_n$ is defined. This support region is divided into non-overlapped sub-regions G_{11}, G_{12}, G_{13} where the gradient vectors f of the image are partitioned into $f_{d_\perp}$ in direction $d_\perp$ and into f_{d_L} in d_L.

For every sub-region the sums $V_{ij}^1, V_{ij}^2, V_{ij}^3, V_{ij}^4$ of the partitioned gradient vectors in the directions $-d_L, +d_L, -d_\perp, +d_\perp$ are calculated and combined to V_{ij}. For three sub-regions and a line with n pixel length, a gradient matrix GM (Eq.1) can be created.

$$GM(L) = \begin{pmatrix} V_{11} & V_{12} & \dots & V_{1n} \\ V_{21} & V_{22} & \dots & V_{2n} \\ V_{31} & V_{32} & \dots & V_{3n} \end{pmatrix} \tag{1}$$

Roman Next we compute the mean vector and the standard deviation vector of GM column vectors:

$$\text{mean}(GM(L)) = \text{mean}\{[V_{11}, V_{12}, V_{13}]^T, [V_{21}, V_{22}, V_{23}]^T, \dots\} \tag{2}$$

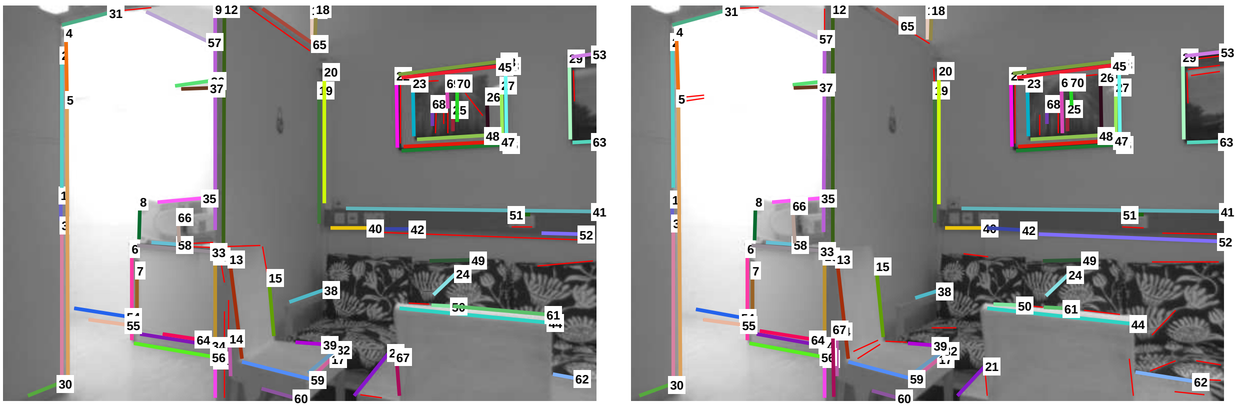
$$\text{std}(GM(L)) = \text{std}\{[V_{11}, V_{12}, V_{13}]^T, [V_{21}, V_{22}, V_{23}]^T, \dots\} \tag{3}$$

After normalization we finally get the $MSLD = [norm(Mean) \;\; norm(Std)]^T$ as a vector with a dimension of $1 \times (3 * 8)$ (for three sub-regions). With the descriptor we can use the euclidean distance as a measurement of similarity of lines. the smaller the distance, the better is the matching of two lines.

Additionally we have used the following geometric constraints:

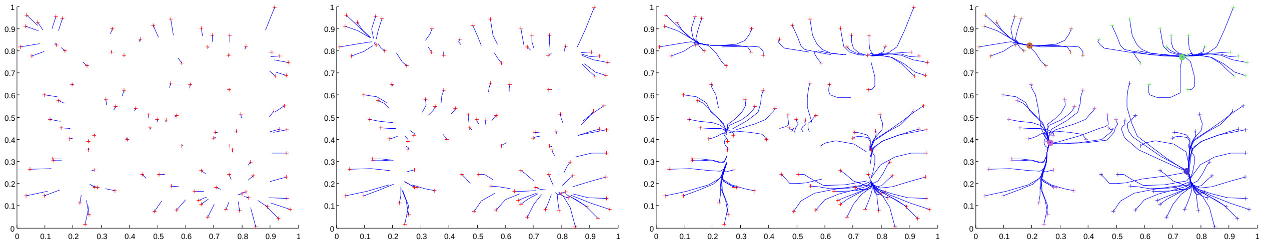
- difference between angle of line in the left and right image must be smaller then threshold θ_d
- disparity between lines must be smaller than threshold d_d
- vertical distance must be smaller than threshold v_d

With all of this constraints the minimum distance of the MSLD's between left and right lines gives us the association for the matching. With the knowledge of the relation, the lines can be mapped in 3D space (according to [3]). Figure 2 shows an example for matching lines and mapping them. In 2(a) and 2(b) the edges of the passage in the left half of the image are fragmented and the edge decomposes into pieces.



(a) left rectified image (b) right rectified image

Figure 2. line matching for two images from a stereo camera setup



(a) 1st iteration (b) 2nd iteration (c) 5th iteration (d) 8th iteration

Figure 3. mean-shift algorithm for 100 random data points and a gaussian kernel with $\lambda = 0.3$ (red points are data points, blue lines are the trajectories)

4. Mean Shift Clustering with 3D line segments

For the merging of lines in state space our approach uses a mean-shift algorithm, first described in [2] and later used in Computer Vision by [1]. The algorithm iteratively shifts every data point to the mean of all data points in defined neighborhood. The advantage of this algorithm is that it is non-parametric, this means that no number of clusters need to be determined (compared to k-means clustering [7]) Beginning with data points $p \in P \subset \mathbb{R}^n$, the sample mean shift $m(x)$ is defined as

$$m(x) = \frac{\sum_P K(p-x)p}{\sum_P K(p-x)} \tag{4}$$

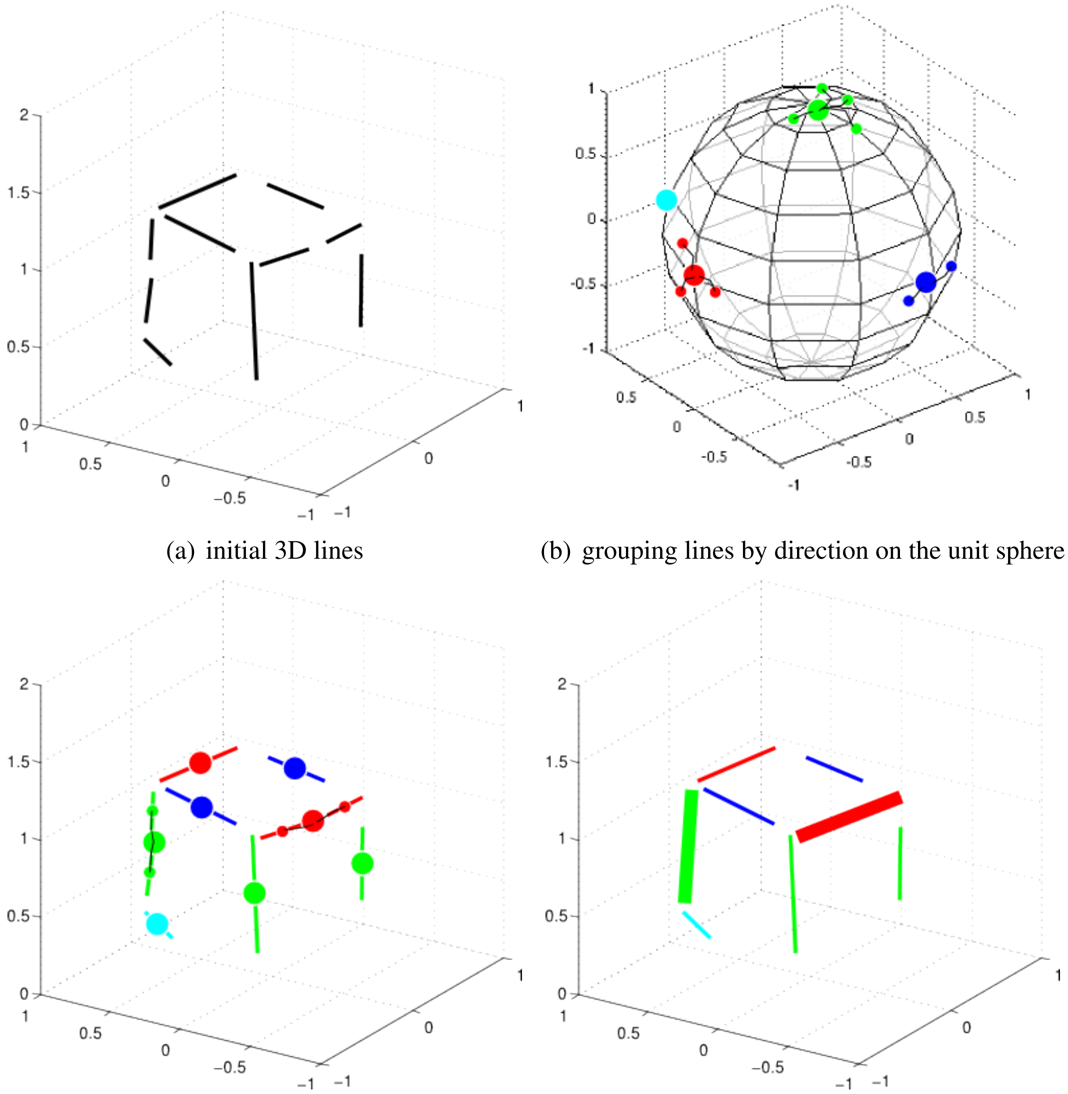
The difference $m(x) - x$ is called *mean shift*. In our work we used a truncated gaussian kernel $K(x)$

$$K(x) = \begin{cases} e^{-\frac{\|x\|^2}{\sigma^2}} & if\ \|x\| \leq \lambda \\ 0 & if\ \|x\| > \lambda \end{cases} \tag{5}$$

Every iteration shifts each data point to its mean, and in the next iteration this progress keeps on until a defined number of iterations or until the maximum mean shift is smaller than a threshold (e.g. $\lambda/1000$). Figure 3 shows the single iterative steps from the mean-shift algorithm based on 100 randomized 2D data points and a gaussian kernel with $\lambda = 0.3$. Finally, in Figure 3(d) 4 clusters are recognizable.

Back to line segments in 3D space (e.g. from our stereo camera setup), based on Lakaemper's work [5], we merge the fragmented line segments in two phases: in the first phase the directions of the lines are clustered in order to find line segments with the same direction. In the second phase the lines with the same direction are merged using an adapted mean-shift algorithm.

For explanation, in Figure 4(a) lines in 3D space without prior knowledge are shown. In the first phase every direction of a line is represented by two points on the unit sphere, the lower half of the unit sphere is a mirror image of the upper half (in Figure 4(b) the mirrored points are omitted for intelligibility). This is important because without mirroring, the data points near the equator would not be grouped adequately and would degrade to two different direction-clusters. The mean-shift-algorithm clusters the direction-points to groups of lines with the same direction. The kernel-size influences the peculiarity of the representing directions, bigger kernel lead to smaller variance of directions. Figure 4(b) shows the distinction of the direction-clusters. In man-made environments the principal directions of the orthonormal isometry converge in visual-compass-like three orthonormal clusters.



(a) initial 3D lines

(b) grouping lines by direction on the unit sphere

(c) clustering lines with centerpoints of lines

(d) combined lines from previous found clusters

Figure 4. grouping lines with mean-shift algorithm: phase 1) grouping directions, phase 2) clustering center points

In the second phase lines with the same direction (all lines with the same direction-cluster from phase 1) are grouped by their center points (see Fig. 4(c)). Phase 1 divides lines in clusters with parallel (resp. semi-parallel) lines, phase 2 aims to sub-cluster them. Only lines with equal directions are merged. Lines which are not broken apart, are grouped in distinct clusters with only one cluster member. In clusters with more than one member the final lines are only determined by these members. The center point is given through the cluster's mean point, the direction is calculated by a weighted mean of the direction of the members and the length follows from the farthermost endpoints (fig. 4(d)). Depending on the kernelsize in the second step parallel lines which are close to each other are also grouped together to build only one representation.

5. Results

This section illustrates the results with the example of a simple door detection. Starting with our stereo vision system, two calibrated cameras with a vertical baseline, the first preprocessing step is to

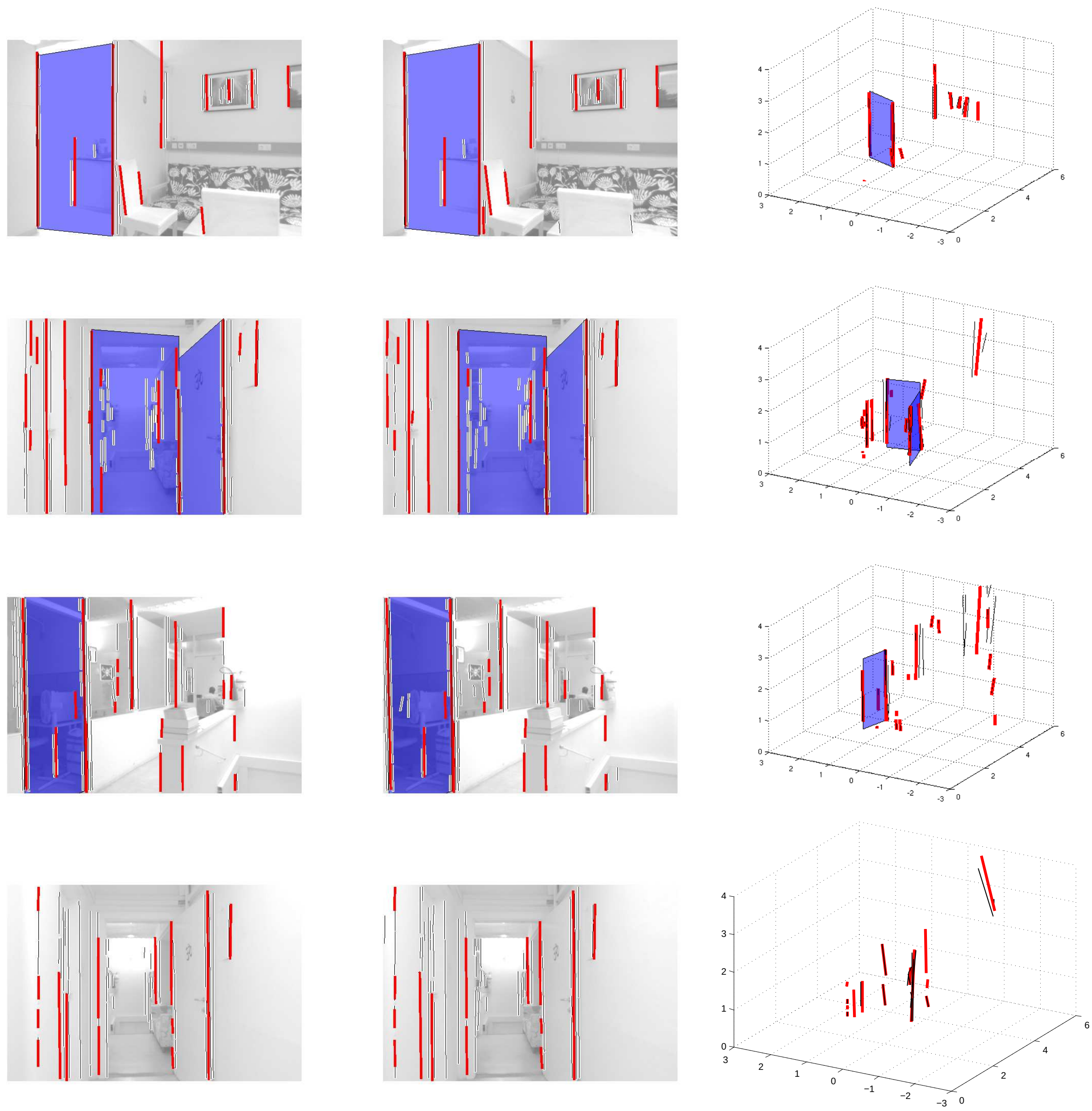


Figure 5. results for door detection by determining two parallel lines with additional constraints: first three rows show successful detection, last row shows false line matching with false grouping

rectify the image with the camera calibration. In the rectified image we use the line detector from [4] to get the line segments. Broken lines caused by overlaps are not especially considered, because the mean-shift clustering (see Section 4.) allows to merge lines after 3D mapping of the matched lines. This is beneficial because no further preprocessing must be done in the 2D images. The mapping process sticks to Hartley and Zisserman [3], whereby the planes are determined by the lines from the two images and the intersection of these planes defines the line in 3D. The grouping algorithm has only the kernel as parameter, so after generating the kernel, whose size is depending on the distance, no other parameter must be set.

Doors are detected where two vertical parallel lines have a distance between 0.6 m and 1.4 m, and both lines have a height of more than 1.5 m. The data set consists of a test run through our office where several doors can be found. Figure 5 shows example frames: in the first three rows door hypothesis are found and drawn as blue rectangles. The last row shows false grouping and false matching of lines, so the segmented line can not be mapped and grouped in 3D and no door can be found.

6. Conclusion

We presented a line merging technique using a mean shift algorithm. Starting from segments we merged lines only from post-processing data and without manipulating line segments in image space. We built simple door hypothesis with two parallel lines for evaluation and showed that the detection of doors worked adequately.
Future work will demonstrate whether or not the utilization of motion data, and the accumulation of the line segments from the last frames will lead to improvements. Detecting closures by parallel clustered lines will also be interesting for future research.

References

[1] Yizong Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799, Aug 1995.

[2] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, Jan 1975.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[4] P. Kahn, L. Kitchen, and E.M. Riseman. A fast line finder for vision-guided robot navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:1098–1102, 1990.

[5] R. Lakaemper. Simultaneous multi-line-segment merging for robot mapping using mean shift clustering. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1654–1660, Oct. 2009.

[6] D.G. Lowe. Object recognition from local scale-invariant features. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 2:1150 –1157 vol.2, 1999.

[7] J. MacQueen. Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. Math. Stat. Probab., Univ. Calif. 1965/66, 1, 281-297 (1967)., 1967.

[8] S. El Mejdani, R. Egli, and F. Dubeau. Old and new straight-line detectors: Description and comparison. *Pattern Recognition*, 41(6):1845 – 1866, 2008.

[9] P. Neubert, P. Protzel, T. Vidal-Calleja, and S. Lacroix. A fast visual line segment tracker. In *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*, pages 353–360, Sept. 2008.

[10] C. Schmid and A. Zisserman. Automatic line matching across views. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 666–671, Jun 1997.

[11] Joan Solà, Teresa A. Vidal-Calleja, and Michel Devy. Undelayed initialization of line segments in monocular slam. In *IROS*, pages 1553–1558, 2009.

[12] Zhiheng Wang, Fuchao Wu, and Zhanyi Hu. Msld: A robust descriptor for line matching. *Pattern Recognition*, 42(5):941 – 953, 2009.